

# Complete Guide: Setting Up AdGuard Home on Xiaomi BE7000 Router

This guide details the process of installing AdGuard Home via Docker on a Xiaomi BE7000 router, resolving the common port 53 conflict with the built-in dnsmasq service, configuring DNS forwarding correctly, and fixing the router's internet status LED indication.

**Disclaimer:** Modifying your router's system files via SSH involves risks. Incorrect commands can lead to instability or require a factory reset. Proceed carefully, understand each command, and ensure you have backups if possible.

## Prerequisites

1. Xiaomi BE7000 Router.
2. AdGuard Home Docker container installed (e.g., via Simple Docker or another method) and accessible via its web interface initially (e.g., <http://192.168.31.1:3000> or the port you chose).
3. A computer on the same network (LAN/Wi-Fi) as the router.
4. An SSH client installed on your computer (e.g., PuTTY for Windows, Terminal for macOS/Linux).
5. Router IP Address: Assumed to be 192.168.31.1. Adjust commands if yours is different.

## Step 1: Gain Root SSH Access

Root access is required to modify system configurations.

1. Use the xmir-patcher tool: <https://github.com/openwrt-xiaomi/xmir-patcher>
2. **Carefully follow all instructions** provided by the xmir-patcher project to enable root SSH access on your specific router firmware version.
3. Ensure you obtain or set the root password during this process.

## Step 2: Connect to Router via SSH

1. Open your SSH client.
2. Connect using the root account and specify the required RSA algorithm:  
`ssh -o HostKeyAlgorithms=ssh-rsa root@192.168.31.1`
3. **First-time connection / Host Key Changed Warning:** If you see a "WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!" message, it's likely because the router's SSH key changed after patching. Remove the old key from your

computer:

- On Windows (PowerShell/CMD): `ssh-keygen -R 192.168.31.1`
- On macOS/Linux (Terminal): `ssh-keygen -R 192.168.31.1`
- Then, try connecting again via SSH. When prompted "Are you sure you want to continue connecting (yes/no/[fingerprint])?", type yes and press Enter.

4. Enter the root password when prompted.

### Step 3: Free Up Port 53 (Change dnsmasq Port)

Move the router's default dnsmasq service off port 53 so AdGuard Home can use it.

1. In the SSH session, run these commands:

```
# Set the dnsmasq listening port to 54
uci set dhcp.@dnsmasq[0].port='54'
```

```
# Save the changes
uci commit dhcp
```

```
# Restart the dnsmasq service
/etc/init.d/dnsmasq restart
```

*(Note: You might see harmless "cp: can't stat..." or "sh: out of range" errors here, but the port change should still apply).*

2. **Verify (Optional):** Check if dnsmasq is listening on port 54:  
`netstat -lnp | grep dnsmasq`

(Look for entries like `*:54` or `192.168.31.1:54`).

### Step 4: Configure AdGuard Home

1. Access the AdGuard Home web setup interface (e.g., `http://192.168.31.1:3000` or the port you exposed for the web UI).
2. Proceed through the setup wizard.
3. **Crucially:** When prompted for the **Admin Web Interface**, choose a port (e.g., 5353 or 8080, *not* 53 or 80). Set the **Listen Interface** to your router's IP (192.168.31.1).
4. When prompted for the **DNS Server**, set the **Listen Interface** to your router's IP (192.168.31.1) and the **Port** to **53**. This should now succeed as port 53 is free.
5. Complete the setup, configuring upstream DNS servers *within AdGuard Home* (e.g., Cloudflare 1.1.1.1, Google 8.8.8.8, Quad9 9.9.9.9, or others).

### Step 5: Configure Router DNS Forwarding (DHCP)

Tell devices on your network to use AdGuard Home for DNS.

1. In the SSH session, run these commands:

```
# Remove any existing DNS server options pushed via DHCP
while uci -q delete dhcp.lan.dhcp_option; do ;; done
```

```
# Add DHCP option 6 (DNS Server) pointing clients to the router's IP (AdGuard Home)
```

```
uci add_list dhcp.lan.dhcp_option='6,192.168.31.1'
```

```
# Save the changes
```

```
uci commit dhcp
```

```
# Restart the dnsmasq service (handles DHCP)
```

```
/etc/init.d/dnsmasq restart
```

## Step 6: Fix Router Internet Status LED (Red LED Issue)

Configure dnsmasq (on port 54) to forward its own DNS checks to AdGuard Home (on port 53).

1. In the SSH session, run these commands:

```
# Tell dnsmasq to NOT use ISP's DNS servers
```

```
uci set dhcp.@dnsmasq[0].noresolv='1'
```

```
# Clear any existing upstream DNS server list for dnsmasq
```

```
while uci -q delete dhcp.@dnsmasq[0].server; do ;; done
```

```
# Add AdGuard Home (localhost port 53) as the upstream server for dnsmasq
```

```
uci add_list dhcp.@dnsmasq[0].server='127.0.0.1#53'
```

```
# Save the changes
```

```
uci commit dhcp
```

```
# Restart dnsmasq
```

```
/etc/init.d/dnsmasq restart
```

2. Wait a minute or two. If the LED doesn't turn white, try rebooting the router completely: reboot

## Step 7: Verification

1. **Renew DHCP Lease:** On your computer/devices, disconnect and reconnect to Wi-Fi, restart network interfaces, or reboot them to get the new DNS settings.
2. **Check Device DNS:** Confirm devices are using 192.168.31.1 as their DNS server.
3. **Check AdGuard Home:** Open the AdGuard Home web UI (e.g., <http://192.168.31.1:5353>). Check the Query Log to see requests from your devices. Test ad blocking.
4. **Check Router Status:** Ensure the internet LED is white and internet access works correctly. Check the router's WAN settings in its web UI - ensure it's set to get DNS automatically from the ISP (recommended) or uses reliable public DNS servers, *not* 192.168.31.1.

You should now have a functional AdGuard Home setup handling DNS for your network, with the port conflict resolved and the router status LED showing correctly.